ACSIJ Advances in Computer Science: an International Journal, Vol. 4, Issue 2, No.14 , March 2015
ISSN : 2322-5157
www.ACSIJ.org

ACSIJ
WWW.ACSIJ.ORG

Finding the Most Influential User of a Specific Topic on the Social Networks

Nghe Nguyen[1], Thanh Ho[2] and Phuc Do[1]

[1]University of Information Technology, Vietnam National University
Ho Chi Minh City, Vietnam
nxnghe@gmail.com, phucdo@uit.edu.vn

[2]Faculty of Information System, University of Economics and Law
VNU-HCM, Vietnam
thanhht@uel.edu.vn

Abstract

The Study on "Maximizing the Spread of Influence through a Social Network" has been strongly attracted attentions recently. One of the most important problems is figuring those who are able to make the strongest influence on the others in spreading information based on a specific topic. We would like to build a system to support viral marketing on social networks and to promote the topic modeling, the propagation model and propagation algorithm to find out the most influential group of users of each topic. The system consists of several steps, such as word extracting, data processing and finding the most influential users in exchanged topics. We especially focus on calculating the users' influence probabilities via an action log file, using the propagation model – TLT (Topic-aware Linear Threshold) and the propagation algorithm – CELF (Cost Effective Lazy Forward). Furthermore, we also experiment our model with Enron email data, which includes 11,177 emails exchanged among 147 users and estimated in 50 topics. We have received many useful topics and the most influential group of users.

Keywords: *influence spread, seed users, topic modeling, viral marketing.*

1. Introduction

Nowadays, the exchange of information through the social networks like Facebook, Twitter, ...is popular and attracts a lot of users worldwide. Because of the advantages of the connection and sharing, social network is used for the viral marketing [12] on the Internet. Social networks will be a great "marketing environment" to present the information of products to customers and also, it could be "the killer" of the company when the unfavorable news spreads beyond control. Therefore, learning about the social networks, focusing on the target customer groups to understand their needs and find the group of users to spread the beneficial information is a significant problem. There are many research works on "Maximizing the Spread of Influence through a Social Network" [1], [2], [3], [8], [10], [17] and finding the most influential users to spread the information of a specific topic [7], [14]. We would like to build a supporting system for the viral marketing through social networks by solving the problem with the input is the exchanged data, and the output is discussed topics on social networks and the group of users has the ability to spread the information of each topic. Solving this problem, we used the topic modeling – LDA (Latent Dirichlet Allocation), the propagation model – TLT (Topic-aware Linear Threshold) [14] and the propagation algorithm – CELF (Cost Effective Lazy Forward) for finding groups of users to spread most of information of a particular topic. After doing the data cleaning process, we use the LDA model to find two matrices: the distribution matrix of vocabularies-topics and the distribution matrix of messages-topics. Next, we use the K-means algorithm to group messages into group of actions as an action log file for calculating the probability of influenced users. Finally, we apply the propagation model – TLT (Topic-aware Linear Threshold) combine with the CELF algorithm. The paper is organized as 1) Introduction 2) Related Works 3) Finding the most influential group of users of a specific topic 4) Experiments and discussion 5) Conclusion and future works.

2. Related works

2.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a generative probabilistic model for collections of discrete text data. In general, LDA is a three-level hierarchical Bayesian model [5], [13] in which each document is described as a random mixture over a latent set of topics. Each topic is modeled as a discrete distribution of a set of words. LDA is suitable for the set of corpus and the set of grouped discrete data. LDA can be used for modeling the document on the purpose of detecting some underlying topics of that document. The generative process of a set of documents consists of three steps: (i) each document has a probabilistic distribution of its topics; this distribution is estimated as the Dirichlet distribution. (ii) for each word in a document, a specific

topic based on the distribution of the topics of that document is chosen (iii) each keyword will be chosen from the multinomial distribution of the keywords according to the chosen topic.

The purpose of LDA is to detect each word belonging to a specific topic. From that we can guess the label of that topic. The importance of topic model is the posterior distribution. This can be seen as the generative process and the posterior inference for the latent set of variables, which are the keywords of the topic. In LDA, this process is calculated by the equation:

$$p(\theta, \emptyset, \boldsymbol{z}|\boldsymbol{w}, \alpha, \beta) = \frac{p(\theta, \emptyset, \boldsymbol{z}, \boldsymbol{w}|\alpha, \beta)}{p(\boldsymbol{w}|\alpha, \beta)} \quad (1)$$

In the equation (1), we have the variables z, θ, ϕ. For each θ_j which is a vector of topics of document j, z_i is the topic of word w_i, $\phi^{(k)}$ is the matrix KxV with $\phi_{i,j} = p(w_i | z_j)$

However, in equation (1), we cannot precisely calculate with the normal factor $p(w|\alpha, \beta)$. Therefore, we normally use Gibbs Sampling (Griffiths & Steyvers, 2004; Steyvers et al., 2004; Rosen-Z vi et al., 2004) for inference.

2.2 Gibbs Sampling

Gibbs Sampling is a member of a family of algorithms from the Markov Chain Monte Carlo (MCMC) [6]. The MCMC algorithms aim to construct a Markov chain that has the target posterior distribution as its stationary distribution. In other words, after a number of iterations through the chain, sampling from the distribution should converge to be close to sampling from the desired posterior. Gibbs Sampling [16], [18] is based on sampling from conditional distributions of the variables of the posterior. For example, to sample x from the joint distribution $p(x) = p(x_1, x_2, \ldots, x_m)$. We do not have any proper solution to compute p(x), but a representation for the conditional distribution is possible, using Gibbs Sampling would perform the following steps:

1. Randomly initialize each x_i
2. For t=1…T:
   2.1. $x_1^{t+1} \sim p(x_1 | x_2^{(t)}, x_3^{(t)}, \ldots, x_m^{(t)})$
   2.2. $x_2^{t+1} \sim p(x_2 | x_1^{(t)}, x_3^{(t)}, \ldots, x_m^{(t)})$
   2.3. $x_m^{t+1} \sim p(x_m | x_1^{(t)}, x_2^{(t)}, \ldots, x_{m-1}^{(t)})$

This procedure is repeated a number of times until the samples converge to what would be sampled from the true distribution. While convergence is theoretically guaranteed with Gibbs Sampling, there is no way of knowing how many iterations are required to reach the stationary distribution. Therefore, diagnosing convergence is a real problem to the Gibbs Sampling approximate inference method. However, it is fairly good performance. Typically, an acceptable estimation of convergence can be obtained by calculating the log-likelihood or even, in some situations, by inspection of the posteriors.

2.3 Clustering messages using K-Means algorithm

After using the LDA model to figure out the topics, we use K-means algorithm [11] to group the messages into cluster. We consider each cluster as an action and keep it in an action log file and calculate the influence probability of the users based on their action. On principle, there are *n* objects, each object has its *m* attributes, and we divide them into *k* groups based on their attributes by using the K-means algorithm.

In our experiment, after running LDA model with Enron email data, which discovers the latent topics and assigns label to latent topics, we use the K-means algorithm to group those messages and convert them into actions. In Table 1, we show 3 actions. They are *a1*, *a2*, and *a3* with their corresponding messages. We have action *a1* means " contract of the power market", action *a2* means "meeting of the power market", action *a3* means “talks about the price of the power market". These actions will be combined with matrix *message - sender - recipient - time* to calculate the influence probability of users by using the Bernoulli distribution model.

2.4 Maximizing the Spread of Influence through a Social Networks

In the problem of Maximizing the Spread of Influence through a Social Network, the input is a social network represented by a directed graph with users as nodes, edges corresponding to social ties, edge weights capturing influence probabilities and a budget *k*. The goal is to find a set of *k* users that the expected influence spread (defined as the expected number of influenced users) is maximized.

Calculating the influence probability: By observation, we have realized that in the social network, there is no existing of available influence probability. We discover a way to calculate the influence probability by using an action log made by each user. Informally, the action log is a table that chronicles any actions performed by every user. Then if we analyze this action log file combine with social network, we can possibly calculate the probability of the most influential users on their friends.

Table 1: Grouping the messages into action with the K-means algorithm

Action	Id	Message's Content
a1 "contract of the power market"	5	[Enron, power, service, contract, Jeff, list, issues, part, contracts, make, plant, supply, First, fuel, opportunities, thing, client, decisions, lawyers]
	6	[Enron, contact, review, contract, copy, offer, distribution, error, party, West, case, plans, basis, anyone, others, copies, reply, National, sender]
a2 "meeting of the power market"	1	[Energy, power, market, time, business, State, week, year, meeting, president, access, utilities, Commission, request, October, part, date, percent, June, time, rates, contracts]
	4	[Energy, Power, information, market, California, thanks, state, meeting, time, electricity, group, people, utilities, Commission, change, markets, Davis, times, June, issue, exchange, investment]
a3 "talks about the price of the power market"	2	[Power, time, agreement, services, credit, Jeff, John, work, customers, issues, comments, Wednesday, deal, prices]
	3	[Energy, power, California, State, electricity, deal, prices, companies, System, President, people, John, office, rate, customers]

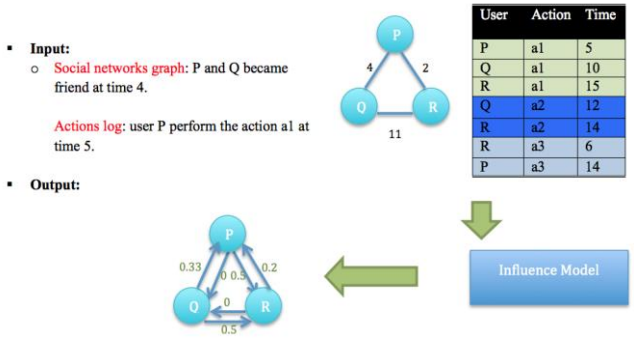

Fig. 1. The influence probability calculation process from the action log file on social networks.

In Fig. 1, we have a social graph as an undirected graph (V, E, T), where V is a set of users. An undirected edge $(v, u) \in E$ between user u and v represents a social tie. We also have an action log **Actions (User, Action, Time)**, which contains a tuple (u, a, t_u) indicating that user u performed action a at time t_u.

A is denoted as the universe of actions, A_u is the number of actions performed by u, $A_{u\&v}$ are the number of actions performed by both u and v, $A_{u|v}$ is the number of actions implemented by either u or v ($A_{u|v}=A_u+A_v- A_{u\&v}$), A_{v2u} the number of actions that propagates from v to u.

According to Amit Goyal [2], we have following definitions:

Definition 1 (*Action propagation*): We say that an action $a \in A$ propagates from user v_i to v_j iff : *(i)* $(v_i,v_j) \in E$; *(ii)* $\exists (v_i,a,t_i),(v_j,a,t_j) \in A$ with $t_i < t_j$; and *(iii)* $\tau(v_i,v_j) \leq t_i$. When this happen, we write ***prop***$(a,v_i,v_j,\Delta t)$ where $\Delta t = t_j - t_i$.

Notice that there must be a social tie between v_i and v_j, both must have performed the action after the moment in which their social tie was created.

Definition 2 (*Propagation graph*): For each action a, we define a propagation graph $PG(a)=(V(a), E(a))$, as follows $V(a) = \{v \mid \exists\ t : (v, a, t) \in A\}$; there is a directed edge $(v_i \xrightarrow{\Delta t} v_j) \in E(a)$ whenever we have a ***prop*** $(a,v_i,v_j,\Delta t)$.

The propagation graph is formed when the users take the action, with their links is under the direction of propagation. When a user takes an action, we say that he is activated for that action. Furthermore, he can take the influence on his friends who have not activated yet. The ability of influence on the neighbors is considered as the influence probability. In brief, we need to calculate the probability of $p_{v,u}$ and $p_{u,v}$ from the edge $(u, v) \in E$.

Bernoulli distribution: In [2], under this model, any time a user v tries to influence its inactive neighbor u, it has a fixed probability of making u activate. If u activates, it is a successful attempt. Each attempt, which is associated with an action, can be viewed as a Bernoulli trial. The Maximum Likehood Estimator (MLE) of success probability is the ratio of number of successful attempts over the total number of trials. Hence, influence probability of v on u using MLE is estimated as:

$$p_{v,u} = \frac{A_{v2u}}{A_v} \tag{2}$$

Algorithm for calculating influence probability: The input of the algorithm consists of a social network graph and an action log file. This file is sorted on action *id* and each action is arranged in time order. Considering the messages of the action (For example, the action a). If the user v sends a message at the time t_v to user u, then later the user u replies it that belongs to action a at the time $t_u (t_u > t_v)$. That is the user v have tried to make the influence on u based on the action a and have been successful (According to Bernoulli distribution model). The probability v can

make the influence on the action *a* is calculated by the number of times *v* has successfully influenced on *u* which is divided by the total number of messages that *v* sent to *u* before. We calculate the influence probability of *v* on *u*, $pa_{v,u}$, with the following formula:

$$pa_{v,u} = \frac{a_{v2u}}{a_{vSentu}} \quad (3)$$

Where

a_{v2u}: The number of times that *v* influenced *u* for the action *a*.

a_{vSentu}: The total number of messages that *v* sent to *u* for the action *a*.

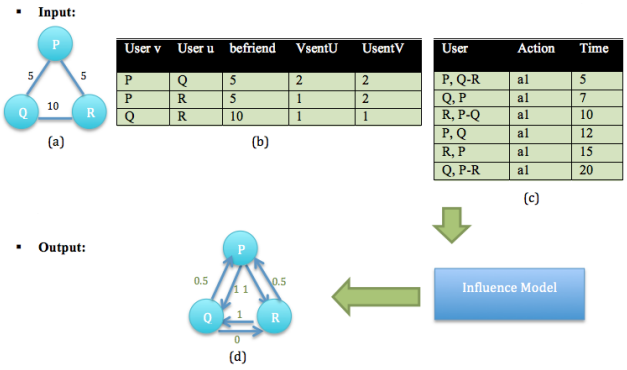

Fig. 2. The influence probability model in this paper

Applying the model of influence probability to equation (3) as above, we propose the following algorithm to read the action log file and calculate the influence probability as below:

Influence Probability Algorithm

1. **for** each action *a* in topic *z* **do**
2. *Friends* map:*(v,u, $t_{befriend}$,vSentu);*
3. *Actions* map:*(v,multimap(a,t_v,$friendlist_v$))*;
4. **for** each user v<*v,multimap(a,t_v,$friendlist_v$)*>in chronological order **do**
5. **for** each user u<*u,multimap(a,t_u,$friendlist_u$)*>belong to *$friendlist_v$* **do**
6. ciTime = 0; *// the current time v influenced u*
7. **if** ((t_u > t_v) && (t_v >= $tbefriend_{v,u}$)) **then**
8. **if** (t_u>ciTime) **then**
9. increment a_{v2u};
10. ciTime = t_u;
11. **end if**
12. **end if**
13. **end for**
14. $pa_{v,u} = \frac{a_{v2u}}{vSentu}$;
15. add*(v, u, $pa_{v,u}$) to result;*
16. **end for**
17. **end for**

Where

- Line 1, loop each action *a* for the topic *z*.
- Line 2, *Friends* map: The map is used for storing each pair of users *(v, u)* are friends of each other, the time when they were friends, and the total number of messages sent to *u* by *v*(*v, u,*$t_{befriend}$, *v*Sent*u*).
- Line 3, *Actions* map: The map is used for storing the messages of an action in a topic and is arranged in chronological order. The *Actions* map has information of *v* that made the actions and a list of *v's* friends who received messages from *v* *(v,multimap(a,t_v,$friendlist_v$))*.
- Line 4-5, of the algorithm is self-explanatory.
- Line 6, the current time when *v* influenced *u*.
- Line 7, make sure that user *u* must be replied the message after the user *v* *(t_u > t_v)*. By the way, these two times must be after the time that *v* and *u* have been friends *(t_u >t_v >= $tbefriend_{v,u}$)*.
- Line 8, make sure that a message can be examined only one time for *v* influences *u*.
- Line 9-10, increase the time of *v* influences *u* and update the latest influence time.
- Line 14-15, calculate the influence probability and store into the *results* variable.

As an example, Fig. 2(a) shows a social graph containing three users P, Q and R with three edges among them. The edges are labeled with timestamps at which the two users became friends. In Fig. 2(b), we have the total number of messages of P sent to Q and Q sent to P. The action log containing action *a1* is presented in Fig. 2(c) and we have user P sent a message of action *a1* to his friends (Q and R) at the time 5. Using the social graph and action log, we can calculate the influence probability of users by our algorithm. In case, we want to calculate $p_{R,P}$ (the probability of R influences on P). In Fig. 2(c), we have R sent to P a message at the time 10 (row 3) then P sent to Q a message at the time 12 (row 4). R sent to P a message again at the time 15 (row 5) and P did not have any actions when he got the message from R. The total number of messages that R sent to P is 2 and the number of times that R influenced P is 1, so we have $p_{R,P}$ = ½ = 0.5. Similarly, we have $p_{P,R}$ = 1, $p_{P,Q}$ = 1, $p_{Q,P}$ = 0.5, $p_{Q,R}$ = 0, $p_{R,Q}$ = 1.

LT-Linear Threshold: LT (Linear threshold) is a propagation model, each node *u* is influenced by each neighbor *v* according to the weight $p_{v,u}$, in which the total of the weights can make influences on *u* are not greater than 1. For each *u*, we selected a random uniform θ_u threshold in the range [0, 1]. At the time *t* that *u* is not activated (*inactive*). At time t + 1, if the total weights of the

neighbors have been activated (*active*) of *u* is greater or equal θ_u, then u will be counted already activating at time t + 1. This process is repeated until there is no node can be active [9].

Maximizing the Spread of Influence through a Social Network: Given a propagation model *m* (Linear Threshold) and an initial seed set $S \in V$, the expected number of active nodes at the end of the process is the expected (influence) spread, denoted by $\sigma_m(S)$. We have following problem (Influence Maximization): Given a directed and edge-weighted social graph *G=(V, E, p)*, a propagation model *m*, and a number *k <= [V]*, find a set $S \in V$, *[S] <= k*, such that $\sigma_m(S)$ is maximum.

We need a propagation algorithm to select group of users with the ability to spread the strongest information. That means, we need to find a user set *S*, which belongs to *V*, *[S]<=k*, such that $\sigma_m(S)$ is maximum.

The first algorithm is the Greedy algorithm proposed by P. Domingos and M. Richardson [15]. However, when applying this algorithm, they faced to unexpected problems: It was a NP-hard problem and the Greedy algorithm that approximated the problem with the ratio of $1 - 1/e - \varepsilon$ for any $\varepsilon > 0$. Leskovec [12], had an idea about the CELF algorithm (Cost Effective Lazy Forward) which had been improved based on Greedy algorithm with the performance speed faster than the old one 700 times.

Greedy Algorithm (*P. Domingos and M. Richardson [15]* proposed):

Input: G, k, σ_m
Output: seed set S
1.S ← ∅;
2.while |S| < k **do**
3. u ← arg $\max_{w \in V \setminus S}$ ($\sigma_m(S + w) - \sigma_m(S)$);
4. S←S + u;
5.end while

The complex step in the Greedy algorithm is in line 3, we select a node which has the largest spreading value $\sigma_m(S+w)-\sigma_m(S)$ in an expectation of finding the user set *S* that has a maximum spread. We propose the CELF algorithm as below based on the idea of Leskovec [12].

CELF Algorithm
Input: G, k, σ_m
Output: seed set S
1.S←∅; M←∅;
2./* First iteration */
3.for each u ∈ V **do**
4. u.mg = σ_m({u});
5. Add u to M in decreasing order of mg;
6.end for
7. /* Celf */
8.examinedUsers ←∅;flag = false;
9.while S.size< k **do**
10. **for** each u ∈ M **do**
11. u = top (root) element in M;
12. Add u to examinedUsers;
13. u.mg = σ_m(S∪ {u}) -σ_m(S);
14. Reinsert u into M in decreasing order of mg;
15. **if** examinedUsers.size == (V.size \ S.size) **then**
16. flag = true;
17. **end if**
18. **if** flag == true **then**
19. v = top (root) element in M; // M is sorted in decreasing order of mg
20. S ← S ∪ {v};
21. M ← M \ {v};
22. examinedUsers.clear();
23. **end if**
24. **end for**
25.end while

The improvement is maintaining a map *M*, with nodes being corresponded to the users in the social network *G*. *M* contains the user *u* formed (*u.mg*). In which, $u.mg = \sigma_m(S \cup \{u\}) - \sigma_m(S)$, is a marginal value (the value of the ability to spread) of *u* and those we are looking for *S* at the current iteration. At the first iteration, the marginal value of each node is calculated and added to the *M* in the order of decreasing marginal value (lines 1-6). In (line 8), we have two temporary variables *examinedUsers* (containing the examined users) and *flag* (a logical variable confirms that we calculated the marginal values for all remain nodes at the current iteration). By then, the next every iteration, takes the first node of *u* in *M*, calculates the marginal value for *u* then inserts it into *M*. The order of the maintained marginal value is set in the order of descending (lines 11-14). Next step, consider the remaining nodes (except the nodes of S) has been recalculated the marginal value yet or not (use the *flag* variable) (lines 15-17). If the variable *flag = true*, takes the first node in *M* that also the node has the largest marginal value for the current iteration and chooses *v* is the next influential user (lines 18-22).

This optimal algorithm is to avoid repeating the calculation of the marginal value of all nodes in iterations (except the first iteration).

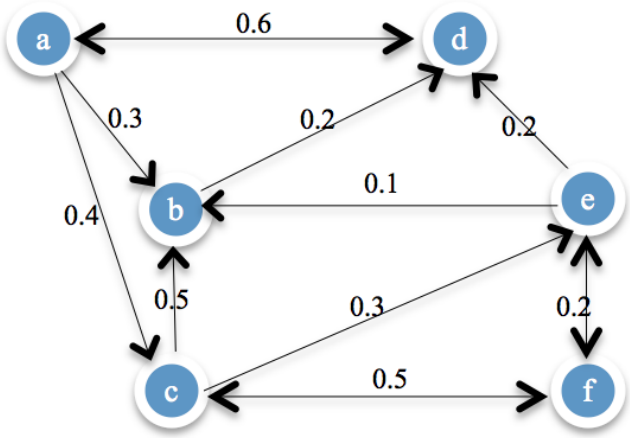

Fig. 3. Finding the influential users group for an action of a topic with our CELF algorithm

As an example, Fig. 3 shows a directed and edge-weighted social graph *G=(V, E, p)* with V is a set of 6 users *a*, *b*, *c*, *d*, *e*, *f* and the probabilities (or weights) *p* of the edges E capturing degrees of influence. In Fig. 3, the probability of the edge (a, b) is 0.3 and it says there is a probability 0.3 with which user *a* influences *b* and thus *a* will propagate to *b* with probability 0.3 for the action. In case, by using the Linear Threshold propagation model with the threshold θ is 0.6, we want to find a set $S \in V$ with k=3 (3 users). We need to find 3 users that $\sigma_m(S)$ (the expected number of active nodes at the end of the process) is maximum. Using our Linear Threshold model and CELF algorithm, we have the result as below (Table 2).

Table 2: Finding the influential users group with Linear Threshold model and CELF algorithm

Steps	S	Results	Notice
1	∅	M:{a=2.0, d=2.0,e=1.0,c=1.0,f =1.0,b=1.0}	The marginal value of each node is added to the map M in the order of decreasing marginal value
2	a	σ_m([a]) = 2.0 M:{a=2.0,d=2.0, e=1.0,c=1.0,f=1.0,b =1.0}	The active nodes are {a,d} is maximum
3	a,f	σ_m([a,f]) = 5.0 M: {f=5.0, c=4.0, b=3.0, e=3.0, d=2.0}	The active nodes are {a,d,f,c,b} is maximum
4	a,f,e **Stop**	σ_m([a,f,e]) = 6.0 M: {e=6.0, d=5.0, b=5.0, c=5.0}	The active nodes are {a ,d,f,c,b,e} is maximum k=3 and the active nodes is 6 users (the total numbers of users).

2.5 Maximizing the Spread of Influence through a Social Networks on a specific topic.

Topic-aware Linear Threshold model (TLT): In [14] For each edge $(v, u) \in E$ and each topic $z \in [1, K]$ we have a probability $p^z_{v,u}$ is the influence of user *v* on user *u* in the topic *z*, so the total weight of influences on each node for each topic cannot be greater than 1. Each node *u* chooses a random threshold θ_u in the range [0, 1]. This model works like the traditional LT model and the influence probability is considered based on specific topics ($p^z_{v,u}$).

Maximizing the Spread of Influence on a specific topic: In [14], Giving a directed social network graph, which has weight for topic z $G_z(V_z,E_z)$, the propagation model m according to current topic $(m=TLT)$, and the number $k <= [V]$, find the user set *S* that belongs to *V*, $[S] <= k$, such that $\sigma_m(S)$ is maximum.

3. Finding the most influential group of users of a specific topic

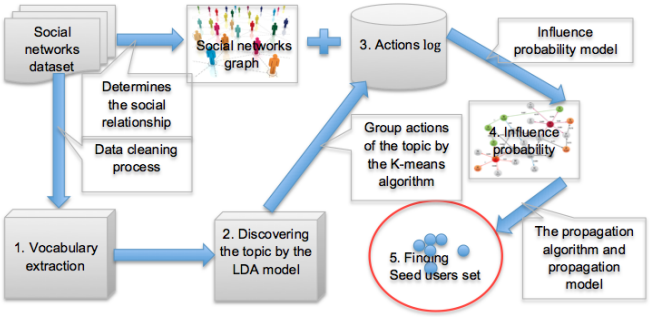

Fig. 4. System architecture

We propose a model shown in Fig. 4. Our model consists of: information extraction, data cleaning, social network analysis to find out the topics by using the LDA model, create the action log file, the influence probability calculation and finding the most influential users. We have the following steps:

Step 1. We do the data cleaning process for the social networks dataset. Each message will be characterized by keywords and removed the stop words...
We create an action log file. For each message of an action of topic we need to identify the sender, a list of recipients and the sending time of the message. We create a matrix *message - sender - recipient – time*.

Step 2. After cleaning the data, by using the LDA model, we will have the matrix words of the topics TxV (word, the distribution probability) and the matrix distributed the messages based on the topics TxD (message's id, the distribution probability).

Step 3. After finding out the topic $z \in [1, K]$ and the distribution of messages by topic *z* from the LDA model, we use the K-means algorithm to group the messages of each topic *z* into actions: *a1*, *a2* and *a3* are the three sub topics of the topic *z* (action's name, message's id).

Step 4. After grouping the actions for the topics, we identify the users that being in friendships of each other and when these relationships were set for the actions. In the paper, two users *u* and *v* have been friend of each other in the action *a* of the topic happened when *v* sent *u* a message and vice versa. Time to send the first message (*v* sent a message to *u* and vice versa) is considered the time that *u* and *v* was be friend for the action *a*.

We continue to use the influence probability model and algorithm were presented in section 2.4 to calculate the influence probability of each user according to specific topics through the actions that have been grouped.

Step 5. After having the result of the influence probability from the step four, we use the TLT (Topic-aware Linear Threshold) model and the propagation algorithm – CELF (Cost Effective Lazy Forward) to find the most influential users groups according to specific topic on social networks.

4. Experiment and discussion

4.1 Input dataset

The dataset is a set of data with 11,177 Enron e-mail messages exchanged between 147 users and estimated in 50 topics.

4.2 Implementation

Vocabulary extraction – data cleaning processing

Each message will be characterized by keywords and removed the stop words... We create a matrix *message - sender - recipient - time* (Table 3).

Table 3: Matrix message - sender - recipient – time

#	Sender	Recipient	Time
1	mike.grigsby@enron.com	k..allen@enron.com	Thu, 27 Dec 2001 07:37:45
1	mike.grigsby@enron.com	frank.ermis@enron.com	Thu, 27 Dec 2001 07:37:45
2	keith.holst@enron.com	k..allen@enron.com	Tue, 23 Oct 2001 12:21:40

Using the LDA model

After cleaning the data, by using the LDA model with the parameters $\alpha = 0.5$, $\beta = 0.1$, the number of iterations for Gibbs sampling is 2000, the number of steps is 100, the number of topics is 50 [16], [18]. We have the matrix words of the topics TxV (word, the distribution probability) and the name of the topics are labeled by hand (Table 4). The matrix distributed the messages based on the topics TxD (message's id, the distribution probability) (Table 5).

Table 4: The distribution of words by the topic

#Topic "contracts"	#Topic "government relationship"	#Topic (power market)
Contract	credit	California
0.053484	0.014981	0.032634
Review	government	market
0.034999	0.014482	0.030124
contact	Cash	electricity
0.030378	0.010989	0.028230
offer	India	prices
0.027077	0.008594	0.027803
…	…	…

Table 5: The distribution of messages by the topic

Messages for the topic "contracts"		#Messages for the topic "government relationship"	
3827	0.947743	4483	0.943262
3832	0.893910	6658	0.925501
8344	0.885986	2003	0.924350
…		…	

K-means algorithm for grouping the actions by the topic

We use the K-means algorithm with the parameters: select a topic, the number messages is 500, the number of groups is 3, and the number of iterations is 10 to group the messages of each topic into actions: a1, a2, a3 are the three sub topics of the topic (action's name, message's id) that have been shown in the Table 6.

Table 6: The distribution of messages for the action by the topic

#action a1 of a topic	#action a2 of a topic	#action a3 of a topic
a1, 8584	a2, 8344	a3, 3827
a1, 3778	a2, 8341	a3, 3832
a1, 8918	a2, 8839	a3, 8844
……	…….	……

Influence probability calculation

After grouping the actions for the topics and determine the users that being in friendships of each other and the time of these relationships were set for the actions. We use the influence probability model and algorithm which were presented in section 2.4 to calculate the influence probability of each user according to specific topics through the actions that have been grouped (Table 7) with the parameters: choose a topic, select the action, the maximum response time of two messages is 15 days.

Table 7: The influence probability by the action of the topic

#The influence probability by the action of the topic#	
sara.shackleton@enron.com, carol.clair@enron.com, 0.67 tana.jones@enron.com, stephanie.panus@enron.com,0 .67 tana.jones@enron.com, susan.bailey@enron.com, 0.67 b..sanders@enron.com, e..haedicke@enron.com, 0.33 j..kean@enron.com, rosalee.fleming@enron.com,0 .50 …	rod.hayslett@enron.com, tracy.geaccone@enron.com,0. 33 james.derrick@enron.com, michelle.cash@enron.com, 0.50 stanley.horton@enron.com, greg.whalley@enron.com, 0.50 marie.heard@enron.com, kim.ward@enron.com, 0.50 sara.shackleton@enron.com, taylor@enron.com, 0.25 …

Finding the most influential users group

After having the result of the influence probability from the Table 7, we use the TLT model combine with CELF algorithm written in java programming language. The input parameters are: choose a topic, select the action, and select the activation threshold = 0.5, k = 5 is the group of users. The final result is the most influential users group for the actions a1, a2, and a3 in one topic. We combine with the information of Enron company's staff and some results from [4], [13] for our result about: employees, topics matching with the employees depending on their positions and their majors. The Table 8 is the most influential users group on the topic "The Contracts". We can see James Derrick was in house lawyer and Jeff Dasovich was government relation executive at Enron…

Table 8: The most influential users group for the topic "The Contracts"

Staffs	Positions
james.derrick@enron.com	In House Lawyer
jeff.dasovich@enron.com	Government Relation Executive
kevin.hyatt@enron.com	Director -Pipeline Business
…	…

Some other topics: The Table 9 shows the most influential users group for the topic "Power Market" and the Table 10 shows the most influential users group for the topic "Government Relations".

Table 9: The most influential users group for the topic "Power Market"

# Topic "Power Market" with LDA Model		
California 0.032634 market 0.030124 electricity 0.028230	Power 0.023683 Price 0.016864 Markets 0.010897	State 0.008955 prices 0.027803 customers 0.008150
The most influential users group with TLT && CELF		
Email	**Position**	
louise.kitchen@enron.com	President-Enron Online	
richard.b.sanders@enron.com	Vice President - Enron WholeSale Services	
richard.shapiro@enron.com	Vice President - Regulatory Affairs	
stanley.horton@enron.com …	President-Enron Gas Pipeline …	

In the "Power Market" topic, we see Louise Kitchen, who was a President of Enron Online. Sanders, who was Vice President of WholeSale Services. Shapiro, who was Vice President of Regulatory Affairs, and Stanley Horton, who was a President of Enron Gas Pipeline. And in the "Government Relations" topic, we see Steffes, who was Vice President of Government Affairs. Keven Presto, who was Vice President. Barry Typcholiz, who was also Vice President and Mark Haedicke, who was Managing Director of Legal Department.

Table 10: The most influential users group for the topic "Government Relations"

# Topic "Government Relations" with LDA model		
Enron 0.026160 credit 0.014981 government 0.014482 Cash 0.010989	Project 0.008394 capital 0.007995 liquidity 0.007895 Company 0.007595	Indian 0.006298 Fimat 0.005699 Dabhol 0.005699 Service 0.005499
The most influential users group with TLT && CELF		
Email	**Position**	
james.d.steffes@enron.com	Vice President - Government Affairs	
kevin.m.presto@enron.com	Vice President	
barry.tycholiz@enron.com	Vice President	
e..haedicke@enron.com	Managing Director - Legal Department	
kevin.m.presto@enron.com …	Trader …	

5. CONCLUSIONS AND FUTURE WORKS

Our research is learning about the propagation models and algorithms through the social networks based on the specific topics. In particular, the research focused on finding the topics with LDA model, document clustering by using K-means algorithm, doing the influence probability calculation, using the propagation model - TLT and the propagation algorithm - CELF. From this research, we have completed the construction of the system with some main steps: extraction and cleaning of data, topic detection, and finding the most influential users group to spread the information of each topic. The test dataset is the Enron email corpus.

With the beginning results, we will probably continue to research and develop the systems with some following directions: Keep going on researching and applying the methods and models to calculate the influence probability. Especially, using the time factor in calculating the influence probability. Developing the system to recommend the most influential users group - "Recommendation System". Whenever a new product is released to the market, we propose *k* groups of users with the ability to spread the product's information in the greatest advantages. Likewise, we also possibly can apply the research in the fields of education, society...The system is also used for the suggesting a profession in any certain field. The issue need to be solved is solving the difficult problem (NP-hard). Furthermore, we should learn and install the algorithms such as Cost - Effective Lazy Forward ++ (CELF ++), Shortest - Path Model (SPM), SIMPATH [1], [2]… or find a new algorithm that can be applied to social networks in reality with the actual number of nodes and the edges that are very large (millions of nodes).

Acknowledgments

This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCM) under grant number B2013-26-02.

References

[1] A.Goyal, W.Lu, and L.V.S.Lakshmanan, "Celf++: optimizing the greedy algorithm for influence maximization in social networks", In 20th International Conference Companion on World Wide Web, WWW 11, New York, NY, USA, 2011.

[2] A.Goyal, "Social Influence and its Applications", A thesis submitted in partial fulfillment of the requirements for the degree of doctor of philosophy in the faculty of graduate studies (Computer Science), The University of British Columbia, 2013.

[3] A.Goyal, Francesco Bonchi, and Laks V. S. Lakshmanan, "A data-Based Approach to Social Influence Maximization", The 38th International Conference on Very Large Data Bases, Istanbul, Turkey, Proceedings of the VLDB Endowment, 2012, Vol. 5, No.1.

[4] Andrew McCallum, Andr´es Corrada, and Xuerui Wang, "The Author-Recipient-Topic Model for Topic and Role Discovery in Social Networks: Experiments with Enron and Academic Email", Department of Computer Science, University of MA, 2004.

[5] David M.Blei, "Latent Dirichlet Allocation", Computer Science Division, University of California, Berkeley, CA, 2003.

[6] B. Walsh, "Markov Chain Monte Carlo and Gibbs Sampling", Lecture Notes for EEB 581, version 26 April 2004.

[7] Chenyi Zhang, Jianling Sun, and Ke Wang, "Information Propagation in Microblog Networks", College of Computer Science, Zhejiang University, China and School of Computing Science, Simon Fraser University, Canada, 2013.

[8] D.Kempe, J.Kleinberg, and E.Tardos, "Maximizing the spread of influence through a social network", In Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 03, pages 137–146, New York, NY, USA, 2003.

[9] David Kempe, "Maximizing the Spread of Influence through a Social Network" slides - pages 5–23, University of Washington Colloquium, 2004.

[10] Francesco Bonchi*, "Influence Propagation in Social Networks: A Data Mining Perspective", 2011.

[11] http://en.wikipedia.org/wiki/K-means_clustering

[12] J.Leskovec, L.A.Adamic, and B.A.Huberman, "The dynamics of viral marketing", In ACM Trans, 2007, volume 1.

[13] Muon Nguyen, Thanh Ho, and Phuc Do, "Social Networks Analysis Based on Topic Modeling", The 10th IEEE RIVF

International Conference on Computing and Communication Technologies (P.119-122), Hanoi, 2013.
[14] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco, "Topic-aware Social Influence Propagation Models", 2012 IEEE 12th International Conference on Data Mining, 2012.
[15] P.Domingos, and M.Richardson, "Mining the network value of customers", In Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 01, pages 57–66, New York, NY, USA, 2001.
[16] Tom Griffiths, "Gibbs Sampling in the Generative Model of Latent Dirichlet Allocation", Gruffydd@psych.stanford.edu, 2004.
[17] WeiChen, Chi Wang, and Yajun Wang, "Scalable Influence Maximization for Prevalent Viral Marketing in Large-Scale Social Networks", KDD'10, Washington, DC, USA, 2010.
[18] William M. Darling, "A Theoretical and Practical Implementation Tutorial on Topic Modeling and Gibbs Sampling", School of Computer Science University of Guelph, 2011.

First Author. MS. Nghe Nguyen graduated from Can Tho University, Can Tho City, Vietnam with a degree in Information Technology. By then, he continues to study master's degree at University of Information Technology, VNU-HCM, Vietnam. Working at Prime Labor Company at the present of Japan as Senior Engineer. His strong ability is about website, mobile and desktop applications with the programming languages as Java, Php, Objective C, Android, and the database as MySQL and Oracle.

Second Author. MS. PhD Student. Thanh Ho works for Faculty of Information System, University of Economics and Law, VNU-HCM, Vietnam. His interests are data mining, e-commerce, Business Intelligent, social network analysis and management information systems. He is a member of Prof. Do Phuc's project.

Third Author. Prof. Do Phuc works for the University of Information Technology, VNU-HCM, Vietnam. His interests are data mining, bioinformatics and social media analysis. His current project is toward the analysis of social network based on the content and structure.